

Тема

«JavaEE»

Состав

Архитектура JavaEE

Архитектура JavaEE

Java EE состоит из набора спецификаций, реализуемых различными контейнерами. Контейнерами называются средства среды времени выполнения Java EE, предоставляющие размещенным на них компонентам определенные службы, например управление жизненным циклом разработки, внедрение зависимости, параллельный доступ и т. д. Такие компоненты используют точно определенные контракты для сообщения с инфраструктурой Java EE и с другими компонентами. Перед развертыванием они должны упаковываться стандартным способом (повторяя структуру определенного каталога, который может быть сжат в архивный файл). Java EE представляет собой расширенный набор функций платформы Java SE, что означает, что API-интерфейсы Java SE могут использоваться любыми компонентами Java EE.

В среде времени выполнения Java EE выделяют четыре типа компонентов, которые должна поддерживать реализация.

- Апплеты представляют собой приложения из графического пользовательского интерфейса (GUI), выполняемые в браузере. Они задействуют насыщенный интерфейс Swing API для производства мощных пользовательских интерфейсов.
- Приложениями называются программы, выполняемые на клиентской стороне. Как правило, они относятся к графическому пользовательскому интерфейсу (GUI) и применяются для пакетной обработки. Приложения имеют доступ ко всем средствам среднего звена.
- Веб-приложения (состоят из сервлетов и их фильтров, слушателей веб-событий, страниц JSP и JSF) выполняются в веб-контейнере и отвечают на запросы HTTP от веб-клиентов. Сервлеты также поддерживают конечные точки веб-служб SOAP и RESTful. Веб-приложения также могут содержать компоненты EJB Lite (подробнее об этом читайте в гл. 7).
- Корпоративные приложения (созданные с помощью технологии Enterprise Java Beans, службы сообщений Java Message Service, интерфейса Java API для транзакций, асинхронных вызовов, службы времени, протоколов RMI-IIOP) выполняются в контейнере EJB. Управляемые контейнером компоненты EJB служат для обработки транзакционной бизнес-логики. Доступ к ним может быть как локальным, так и удаленным по протоколу RMI (или HTTP для веб-служб SOAP и RESTful).

Сервисы Контейнеры развертывают свои компоненты, предоставляя им соответствующие базовые сервисы. Контейнеры позволяют разработчику сконцентрироваться на реализации бизнес-логики, а не решать технические проблемы, присутствующие в корпоративных приложениях. Например, EJB-контейнеры предоставляют коннекторы для доступа к информационной

системе предприятия, но не к контейнеру апплетов или контейнеру клиентского приложения. Java EE предлагает следующие сервисы:

□ Java API для транзакций — этот сервис предлагает интерфейс разграничения транзакций, используемый контейнером и приложением. Он также предоставляет интерфейс между диспетчером транзакций и диспетчером ресурсов на уровне интерфейса драйвера службы. □ Интерфейс сохраняемости Java — стандартный интерфейс для объектно-реляционного отображения (ORM). С помощью встроенного языка запросов JPQL вы можете обращаться к объектам, хранящимся в основной базе данных.

□ Валидация — благодаря валидации компонентов объявляется ограничение целостности на уровне класса и метода.

□ Интерфейс Java для доступа к службам сообщений — позволяет компонентам асинхронно обмениваться данными через сообщения. Он поддерживает надежный обмен сообщениями по принципу «от точки к точке» (P2P), а также модель публикации-подписки (pub-sub).

□ Java-интерфейс каталогов и служб именования (JNDI) — этот интерфейс, появившийся в Java SE, используется как раз для доступа к системам служб именования и каталогов. Ваше приложение применяет его, чтобы ассоциировать (связывать) имена с объектами и затем находить их в каталогах. Вы можете задать поиск источников данных, фабрик классов JMS, компонентов EJB и других ресурсов. Интерфейс JNDI, повсеместно присутствовавший в коде до версии 1.4 J2EE, в настоящее время используется более прозрачным способом — через внедрение.

□ Интерфейс JavaMail — многим приложениям требуется функция отправки сообщений электронной почты, которая может быть реализована благодаря этому интерфейсу.

□ Фреймворк активизации компонентов JavaBeans (JAF) — интерфейс JAF, являющийся составной частью платформы Java SE, предоставляет фреймворк для обработки данных различных MIME-типов. Используется сервисом JavaMail.

□ Обработка XML — большинство компонентов Java EE могут развертываться с помощью опциональных дескрипторов развертывания XML, а приложениям часто приходится манипулировать XML-документами. Интерфейс Java для обработки XML (JAXP) поддерживает синтаксический анализ документов с применением интерфейсов SAX и DOM, а также на языке XSLT.

□ Обработка JSON (объектной нотации JavaScript) — появившийся только в Java EE 7 Java-интерфейс для обработки JSON (JSON-P) позволяет приложениям синтаксически анализировать, генерировать, трансформировать и запрашивать JSON.

□ Архитектура коннектора Java EE — коннекторы позволяют получить доступ к корпоративным информационным системам (EIS) с компонента Java EE. К таким компонентам относятся базы данных, мейнфреймы либо программы для планирования и управления ресурсами предприятия (ERP).

□ Службы безопасности — служба аутентификации и авторизации для платформы Java (JAAS) позволяет сервисам аутентифицироваться и устанавливать права доступа, обязательные для пользователей. Контракт поставщика сервиса авторизации Java для контейнеров (JACC) определяет соглашение о взаимодействии между сервером приложений Java EE и

поставщиком сервиса авторизации, позволяя, таким образом, сторонним поставщикам такого сервиса подключаться к любому продукту Java EE. Интерфейс поставщика сервисов аутентификации Java для контейнеров (JASPI) определяет стандартный интерфейс, с помощью которого модули аутентификации могут быть интегрированы с контейнерами. В результате модули могут установить идентификаторы подлинности, используемые контейнерами.

□ Веб-службы — Java EE поддерживает веб-службы SOAP и RESTful. Интерфейс Java для веб-служб на XML (JAX-WS), сменивший интерфейс Java с поддержкой вызовов удаленных процедур на основе XML (JAX-RPC), обеспечивает работу веб-служб, работающих по протоколу SOAP/HTTP. Интерфейс Java для веб-служб RESTful (JAX-RS) поддерживает веб-службы, использующие стиль REST.

□ Внедрение зависимостей — начиная с Java EE 5, некоторые ресурсы могут внедряться в управляемые компоненты. К таким ресурсам относятся источники данных, фабрики классов JMS, единицы сохраняемости, компоненты EJB и т. д. Кроме того, для этих целей Java EE 7 использует спецификации по контексту и внедрению зависимости (CDI), а также внедрение зависимости для Java (DI).

□ Управление — Java EE с помощью специального управляющего компонента определяет API для операций с контейнерами и серверами. Интерфейс управляющих расширений Java (JMXAPI) также используется для поддержки управления.

□ Развертывание — спецификация Java EE по развертыванию определяет соглашение о взаимодействии между средствами развертывания и продуктами Java EE для стандартизации развертывания приложения.

Упаковка

Для последующего развертывания в контейнере компоненты сначала необходимо упаковать в стандартно отформатированный архив. Java SE определяет файлы архива Java (формат JAR), используемые для агрегации множества файлов (Java-классов, дескрипторов развертывания, ресурсов или внешних библиотек) в один сжатый файл (на основе формата ZIP).

Модуль клиентских приложений содержит Java-классы и другие ресурсные файлы, упакованные в архив JAR. Этот файл может выполняться в среде Java SE или в контейнере клиентского приложения. Как любой другой архивный формат, JAR-файл содержит опциональный каталог META-INF для мета-информации, описывающей архив. Файл META-INF/MANIFEST.MF используется для определения данных, относящихся к расширениям и упаковке. При развертывании в контейнере клиентских приложений соответствующий дескриптор развертывания может быть опционально размещен по адресу META-INF/application-client.xml.

□ Модуль EJB содержит один или несколько компонент-сеансов и/или компонентов, управляемых сообщениями (MDB), упакованных в архив JAR (часто называемый JAR-файл EJB). Он содержит опциональный дескриптор развертывания META-INF/ejb-jar.xml и может развертываться только в контейнере EJB.

□ Модуль веб-приложений содержит сервлеты, страницы JSP и JSF, веб-службы, а также любые другие файлы, имеющие отношение к Сети (страницы

HTML и XHTML, каскадные таблицы стилей (CSS), Java-сценарии, изображения, видео и т. д.). Начиная с Java EE 6 модуль веб-приложения также может содержать компоненты EJB Lite (подмножество интерфейса EJB API, описанное в главе 7). Все эти артефакты упаковываются в архив JAR с расширением WAR (также называемый архивом WAR или веб-архивом). Опциональный веб-дескриптор развертывания определяется в файле WEB-INF/web.xml. Если архив WAR содержит компоненты EJB Lite, то файл WEB-INF/ejb-jar.xml может быть снабжен опциональным дескриптором развертывания. Java-файлы с расширением .class помещаются в каталог WEB-INF/classes, а зависимые архивные JAR-файлы — в каталог WEB-INF/lib.

□ Корпоративный модуль может содержать нуль или более модулей веб-приложений, модулей EJB, а также других общих или внешних библиотек. Они упаковываются в корпоративный архив (файл JAR с расширением .ear) таким образом, чтобы развертывание всех этих модулей происходило одновременно и согласованно. Опциональный дескриптор развертывания корпоративного модуля определяется в файле META-INF/application.xml. Специальный каталог lib используется для разделения общих библиотек по модулям.

Спецификации

Спецификация Версия JSR URL JSF 2.2 344 <http://jcp.org/en/jsr/detail?id=344> JSP 2.3 245 <http://jcp.org/en/jsr/detail?id=245> JSTL 1.2 52 <http://jcp.org/en/jsr/detail?id=52> Servlet 3.1 340 <http://jcp.org/en/jsr/detail?id=340> WebSocket 1.0 356 <http://jcp.org/en/jsr/detail?id=356> Expression Language 3.0 341 <http://jcp.org/en/jsr/detail?id=341> EJB Lite 3.2 345 <http://jcp.org/en/jsr/detail?id=345> JPA 2.1 338 <http://jcp.org/en/jsr/detail?id=338> JTA 1.2 907 <http://jcp.org/en/jsr/detail?id=907> Bean Validation 1.1 349 <http://jcp.org/en/jsr/detail?id=349> Managed Beans 1.0 316 <http://jcp.org/en/jsr/detail?id=316> Interceptors 1.2 318 <http://jcp.org/en/jsr/detail?id=318> Contexts and Dependency Injection 1.1 346 <http://jcp.org/en/jsr/detail?id=346> Dependency Injection for Java 1.0 330 <http://jcp.org/en/jsr/detail?id=330> Debugging Support for Other Languages 1.0 45 <http://jcp.org/en/jsr/detail?id=45> JAX-RS 2.0 339 <http://jcp.org/en/jsr/detail?id=339> JSON-P 1.0 353 <http://jcp.org/en/jsr/detail?id=353>

Материалы для подготовки

1. [https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%BB%D0%B5%D1%82_\(Java\)](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%BB%D0%B5%D1%82_(Java))
2. <https://habrahabr.ru/post/88513/>
3. <https://habrahabr.ru/post/113145/>
4. <https://habrahabr.ru/post/247647/>
5. <http://logging.apache.org/log4j/2.x/manual/>
6. <http://logback.qos.ch/manual/index.html>
7. <http://www.slf4j.org/docs.html>
8. <https://commons.apache.org/proper/commons-logging/>

Вопросы для самоконтроля

- 1) Опишите основные компоненты архитектуры JavaEE
- 2) В чем преимущество определенных фреймворков логгирования?